On Training Bi-directional Neural Network Language Model with Noise Contrastive Estimation

Tianxing He, Yu Zhang, Jasha Droppo, and Kai Yu

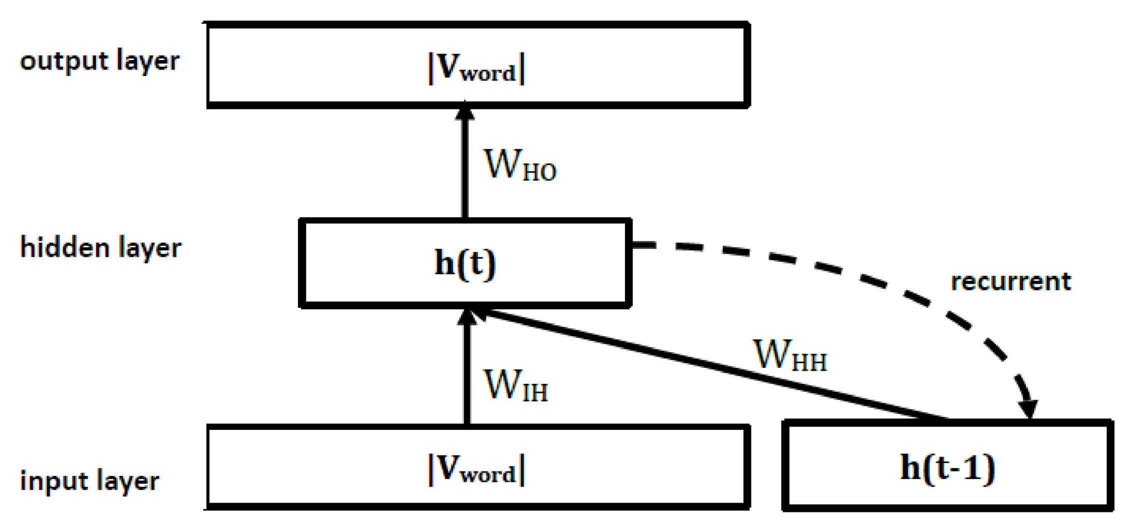
cloudygoose@sjtu.edu.cn,yzhang87@csail.mit.edu, jdroppo@microsoft.com,kai.yu@sjtu.edu.cn

Overview

- **Motivation: MLE is not suitable for training bi-directional neural network language model.**
- **Approach: Use sentence-level NCE to achieve sentence-level normalization.**
- **Experiments&Discussion: Our proposed model performs well on a sanity pseudo PPL check, but unfortunately, it did not out-perform our uni-directional baselines.**

Background: Recurrent Neural Network Language Model

- RNNLM encodes all history with recurrent connections:



Diffculty of Training bi-directional Language Model

The definition of uni-directional lm ensures its sentence-level normalization, which enables us to apply MLE framework.

$$P(\mathcal{W}) = \Pi_i P(w_i|w_{1..i-1})$$

$$\sum_{\mathcal{W}} P_{LM}(\mathcal{W}) = 1$$

However a bi-directional lm doesn't satisfy that condition. For example:

$$P(w_i|w_{1..i-1,i+1..N})$$

Training bi-NNLM with NCE

- (Noise Contrastive Estimation)NCE fits an unnormalized model to the data distribution by learning a normalization constant.

$$\begin{aligned} J_{NCE}(\theta) = & E_{P_{data}(\mathcal{W})}[logP(D=1|\mathcal{W};\theta)] \\ & + kE_{P_{noise}(\mathcal{W})}[logP(D=0|\mathcal{W};\theta)] \end{aligned}$$

$$P(D=1|\mathcal{W};\theta) = \frac{P_\theta^{NCE}(\mathcal{W})}{P_\theta^{NCE}(\mathcal{W}) + kP_{noise}(\mathcal{W})}$$

$$P(D=0|\mathcal{W};\theta) = \frac{kP_{noise}(\mathcal{W})}{P_\theta^{NCE}(\mathcal{W}) + kP_{noise}(\mathcal{W})}$$

Model Formulation

In this work, P(W) consists of the product of word-level scores(similar to uni-directional LM) and a learned normalization scalar c, required by the NCE framework to ensure normalization

$$\mathbf{v}_i = \mathbf{W}_{xh}\mathbf{x}_i \longleftarrow \text{One-hot representation}$$

$$\overrightarrow{\mathbf{h}}_i^1 = g(\overrightarrow{\mathbf{h}}_{i-1}^1, \mathbf{v}_i) \longleftarrow \text{Gated Recurrent Unit(GRU)}$$

$$\overleftarrow{\mathbf{h}}_i^1 = g(\overleftarrow{\mathbf{h}}_{i+1}^1, \mathbf{v}_i)$$

$$\mathbf{h}_i^1 = tanh(\mathbf{W}_{hf}^1 \overrightarrow{\mathbf{h}}_i^1 + \mathbf{W}_{hr}^1 \overleftarrow{\mathbf{h}}_i^1 + \mathbf{b}^1)$$

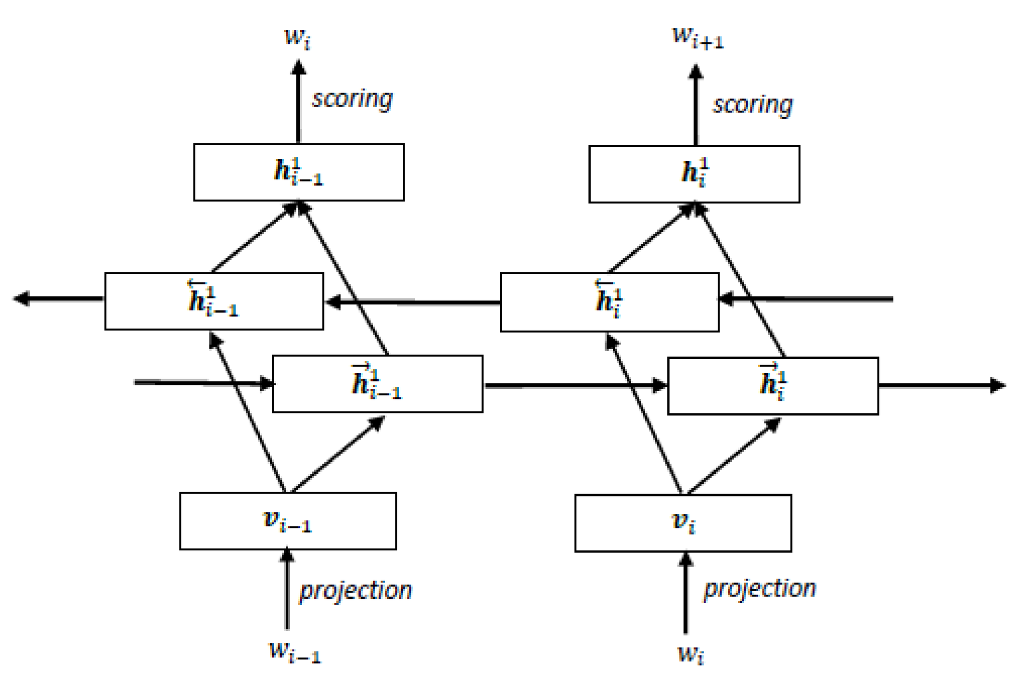
$$\mathbf{u}_i = exp(\mathbf{W}_{ho}\mathbf{h}_i^1 + \mathbf{b}_o)$$

$$f_i(\mathcal{W}) = \frac{\mathbf{u}_i(w_i)}{\sum_{w_j \in V} \mathbf{u}_i(w_j)}$$

$$f'(\mathcal{W}) = \Pi_i f_i(\mathcal{W})$$

$$P^{NCE}(\mathcal{W}) = f'(\mathcal{W})exp(c)$$

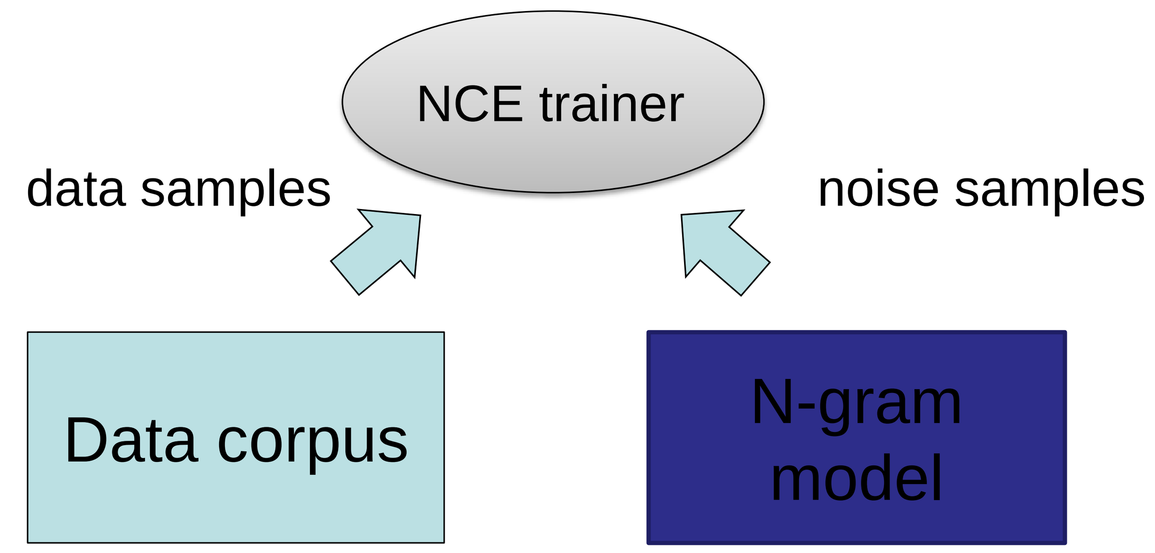
Learned normalization constant



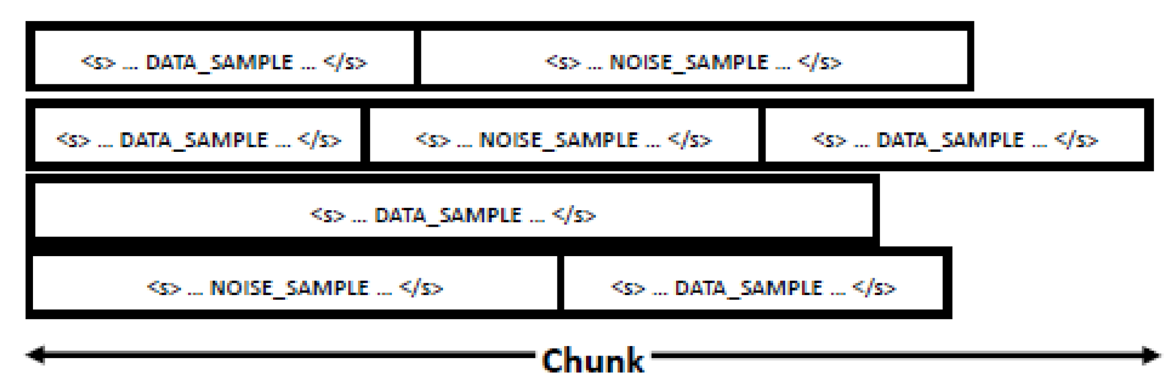
Training&Implementation Details

Stochastic Gradient Descent(SGD) with learning rate(lr) decaying is used.

The SRILM toolkit is used to build N-GRAM models as baselines.



We parallel on sentence-level to utilize the GPU speedup.



Pseudo-PPL Sanity check

We check bi-nnlm's pseudo on different kinds of texts, test-ptb is real data, 4gram-text is samples from a 4-gram model, uniform-text is completely randomly generated sentences.

Model	Pseudo-PPL		
	test-ptb	**4gram-text**	**uniform-text**
UNI-GRULM	103.7	431.0	91935.7
BI-GRULM(MLE)	1.12	1.16	3.358
BI-GRULM(NCE)	15.5	3846.4	99565.4

It's clear that NCE trained BI-GRULM's behavior is more similar to a normalized model.

Experiments on ptb-rescore task

To make our training time tolerable, we designed a task similar to "sentence completion" on the ptb dataset. The models are expected to assign higher sentence-level scores to the original sentence than the distorted sentences:

original	no it was n't black monday
s-error	no it was n't black **revoke**
d-error	no it was n't monday
i-error	no it **cracks** was n't black monday

The accuracy for each model is shown in the table below, in the exploration, we also found a length-norm trick that helps a lot of deletion error:

$$score_{length-norm}(\mathcal{W}) = \frac{score(\mathcal{W})}{l} = \frac{\sum_i^l log f_i(\mathcal{W})}{l}$$

Model	noise ratio	Accuracy(%)/Accuracy after **length-norm**(%)			
		test-s	**test-d**	**test-i**	**test-sdi**
4-GRAM	-	75.4/n75.4	3.2/n12.7	**100**/n**98.2**	13.4/n40.8
UNI-GRULM	-	**80.6**/n**80.6**	**3.9**/n**21.8**	99.9/n96.9	**20.2**/n**60.9**
BI-GRULM(MLE)	-	50.0/n50.0	0.31/n21.9	95.3/n31.5	6.8/n27.1
BI-GRULM (NCE)	1	31.9/n31.9	3.9/n12.8	67.4/n53.0	10.9/n17.8
	10	39.9/n39.9	8.8/n19.4	61.8/n48.8	20.5/n26.2
	20	39.2/n39.2	**11.0**/n**21.6**	59.1/n45.3	**21.0**/n26.3
	50	48.4/n48.4	6.8/n19.8	74.2/n54.9	18.1/n29.0
	100	**55.7**/n**55.7**	0.5/n13.4	**98.6**/n**80.4**	10.3/n**34.5**

We state two major observations:

- The proposed NCE training for bi-directional GRULM out-performs MLE training.
- The performance can only be improved when the amount of noise samples grow exponentially.

Conclusions

Our proposed NCE training for bi-directional NNLM out-performed the MLE trained model, however, it did not outperform the uni-directional baselines. The reason maybe that sentence-level sampling space is too sparse for our sampling to cover.